U3305049

Imesha Piyumali

Assessment 2: Python application programming

Case Study 1 — Campus Cafe Checkout

Understand the Problem

This system will act as a **simple point-of-sale system** for a campus cafe. It will allow customers to view a menu of items. Those should be added to a shopping cart, view the cart and checkout. Then print a receipt. In the checkout, the program should calculate the subtotal, add 10% tax, and in an optional way it will apply a 5% student discount.

- The system might use a dictionary for menu items.
- A list for the cart representation
- A set for the categories was structured around the functions.
- This program should loop until the user chooses to exit. It should be with functions.

Inputs & Outputs

- **Inputs**:
    - Menu option (applied for integer: 1–5). (int)
    - Item name to add (string). (str)
    - If a student discount applies (string: “yes/no”).(y/n)

- **Outputs**:
    - Display in cafe menu. (categories, prices)
    - After items are added, then confirm.
    - Display for the items in the cart.
    - Receipts show the subtotal, total, tax, discount.
    - Categories in the cart.

Algorithm

- Identifying all the dictionaries of menu items (That categories will be ≥6).
- Start the empty cart or list.
- Display the looping main menu options:
    1. Displaying at the menu.
    2. Should add to the items.
    3. View all the cart details.
    4. Check out the item.
    5. Then Exit.

    If the user chooses view cart,

    Display all car items, quantities, costs.

    Show categories by uniquely. Using set

    If the user chooses checkout,

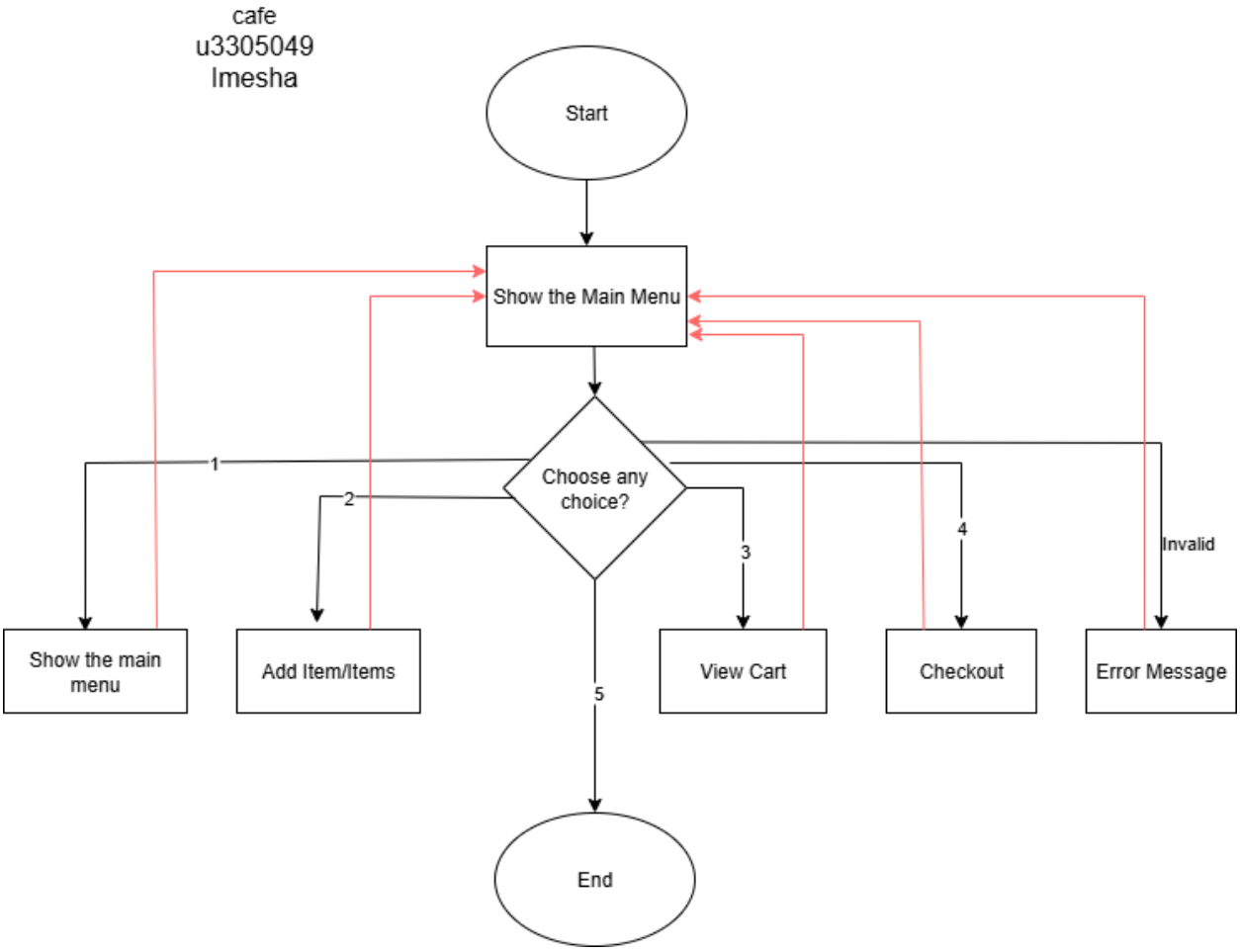
    calculate the subtotal, with calculating the tax (10%), discount will be optional (5%), and calculate the final total.

- Print the well formatted receipt
- Reset the cart to a new order.

Pseudocode

```
BEGIN
  menu = {items with price, category}
  cart = []

  WHILE True
     PRINT main menu
     GET choice
     IF choice == 1: show menu
     IF choice == 2: add item
     IF choice == 3: view cart
     IF choice == 4: checkout
     IF choice == 5: EXIT
     ELSE: invalid choice
  END WHILE
END
```

Flowchart



draw.io link

https://drive.google.com/file/d/1JiU29tff6GoK-nExqpI6A57JnjX0wU3R/view?usp=sharing

Python Code

https://github.com/Imesha-Piyumali/Assessment-2-Python-application-programming

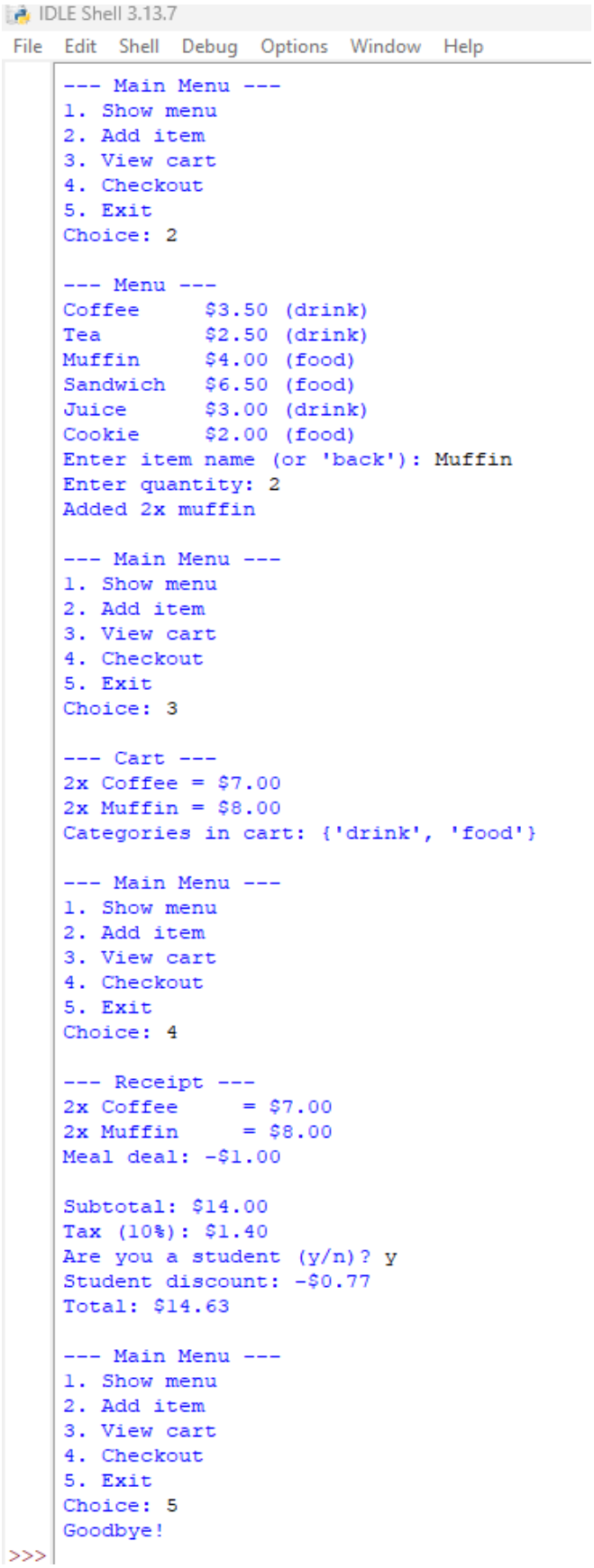
```
# cafe.py
# Student: Imesha Piyumali, ID: u3305049, Date: 17.09.2025

# Campus Café Checkout

MENU = {
    "coffee": (3.50, "drink"),
    "tea": (2.50, "drink"),
    "muffin": (4.00, "food"),
    "sandwich": (6.50, "food"),
    "juice": (3.00, "drink"),
    "cookie": (2.00, "food"),
}

TAX_RATE = 0.10
STUDENT_DISCOUNT = 0.05
MEAL_DEAL_DISCOUNT = 1.00  # $1 off if food+drink present

def show_menu():
    print("\n--- Menu ---")
    for item, (price, category) in MENU.items():
        print(f"{item.capitalize():10s} ${price:.2f} ({category})")

def add_item(cart):
    show_menu()
    choice = input("Enter item name (or 'back'): ").lower()
    if choice in MENU:
        qty = int(input("Enter quantity: "))
        cart.append((choice, qty))
        print(f"Added {qty}x {choice}")
    elif choice != "back":
        print("Item not found.")

def view_cart(cart):
    if not cart:
        print("\nCart is empty.")
        return
    print("\n--- Cart ---")
    for item, qty in cart:
```

```
        price, _ = MENU[item]
        print(f"{qty}x {item.capitalize()} = ${price*qty:.2f}")
    categories = {MENU[item][1] for item, _ in cart}
    print("Categories in cart:", categories)

def checkout(cart):
    if not cart:
        print("Cart is empty.")
        return
    print("\n--- Receipt ---")
    subtotal = 0
    categories = set()
    for item, qty in cart:
        price, category = MENU[item]
        subtotal += price * qty
        categories.add(category)
        print(f"{qty}x {item.capitalize():10s} = ${price*qty:.2f}")
    if "food" in categories and "drink" in categories:
        print(f"Meal deal: -${MEAL_DEAL_DISCOUNT:.2f}")
        subtotal -= MEAL_DEAL_DISCOUNT
    tax = subtotal * TAX_RATE
    total = subtotal + tax
    print(f"\nSubtotal: ${subtotal:.2f}")
    print(f"Tax (10%): ${tax:.2f}")
    if input("Are you a student (y/n)? ").lower() == "y":
        discount = total * STUDENT_DISCOUNT
        total -= discount
        print(f"Student discount: -${discount:.2f}")
    print(f"Total: ${total:.2f}")
    cart.clear()

def main():
    cart = []
    while True:
        print("\n--- Main Menu ---")
        print("1. Show menu")
        print("2. Add item")
        print("3. View cart")
        print("4. Checkout")
        print("5. Exit")
        choice = input("Choice: ")
        if choice == "1":
```

```
            show_menu()
        elif choice == "2":
            add_item(cart)
        elif choice == "3":
            view_cart(cart)
        elif choice == "4":
            checkout(cart)
        elif choice == "5":
            print("Goodbye!")
            break
        else:
            print("Invalid option.")

if __name__ == "__main__":
    main()
```

Test Results:

```
>>> ========== RESTART: D:/UC/IIT/Assignments/Campus Café Checkout.py ==========

--- Main Menu ---
1. Show menu
2. Add item
3. View cart
4. Checkout
5. Exit
Choice: 1

--- Menu ---
Coffee      $3.50 (drink)
Tea         $2.50 (drink)
Muffin      $4.00 (food)
Sandwich    $6.50 (food)
Juice       $3.00 (drink)
Cookie      $2.00 (food)

--- Main Menu ---
1. Show menu
2. Add item
3. View cart
4. Checkout
5. Exit
Choice: 2

--- Menu ---
Coffee      $3.50 (drink)
Tea         $2.50 (drink)
Muffin      $4.00 (food)
Sandwich    $6.50 (food)
Juice       $3.00 (drink)
Cookie      $2.00 (food)
Enter item name (or 'back'): Coffee
Enter quantity: 2
Added 2x coffee

--- Main Menu ---
1. Show menu
2. Add item
3. View cart
4. Checkout
5. Exit
Choice: 2

--- Menu ---
Coffee      $3.50 (drink)
Tea         $2.50 (drink)
Muffin      $4.00 (food)
Sandwich    $6.50 (food)
Juice       $3.00 (drink)
Cookie      $2.00 (food)
```


IDLE Shell 3.13.7
File Edit Shell Debug Options Window Help
--- Main Menu ---
1. Show menu
2. Add item
3. View cart
4. Checkout
5. Exit
Choice: 2
--- Menu ---
Coffee $3.50 (drink)
Tea $2.50 (drink)
Muffin $4.00 (food)
Sandwich $6.50 (food)
Juice $3.00 (drink)
Cookie $2.00 (food)
Enter item name (or 'back'): Muffin
Enter quantity: 2
Added 2x muffin
--- Main Menu ---
1. Show menu
2. Add item
3. View cart
4. Checkout
5. Exit
Choice: 3
--- Cart ---
2x Coffee = $7.00
2x Muffin = $8.00
Categories in cart: {'drink', 'food'}
--- Main Menu ---
1. Show menu
2. Add item
3. View cart
4. Checkout
5. Exit
Choice: 4
--- Receipt ---
2x Coffee = $7.00
2x Muffin = $8.00
Meal deal: -$1.00
Subtotal: $14.00
Tax (10%): $1.40
Are you a student (y/n)? y
Student discount: -$0.77
Total: $14.63
--- Main Menu ---
1. Show menu
2. Add item
3. View cart
4. Checkout
5. Exit
Choice: 5
Goodbye!
>>>

Step 6 — Testing

Test Case 1

Input: Add 2 coffees, 1 muffin, checkout as student
Expected Outcome:1

- Subtotal = (2×3.50 + 4.00) = 11.00
- Meal deal applies: -1.00 → 10.00
- Tax 10% = 1.00 → 11.00
- Student's discount 5% = 0.55 → 10.45
  Expected Outcome: Total = **$10.45**

Test Case 2

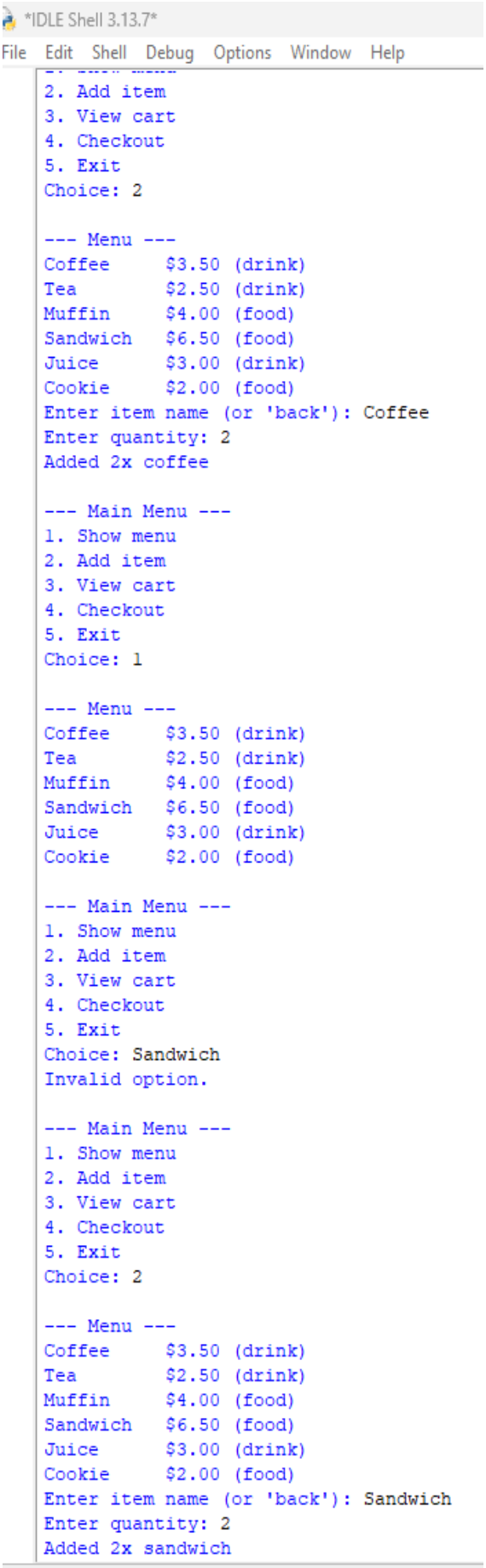
Input: Add 1 tea, checkout as non-student
Expected Outcome: Subtotal = 2.50 → Tax = 0.25 → Total = **$2.75**

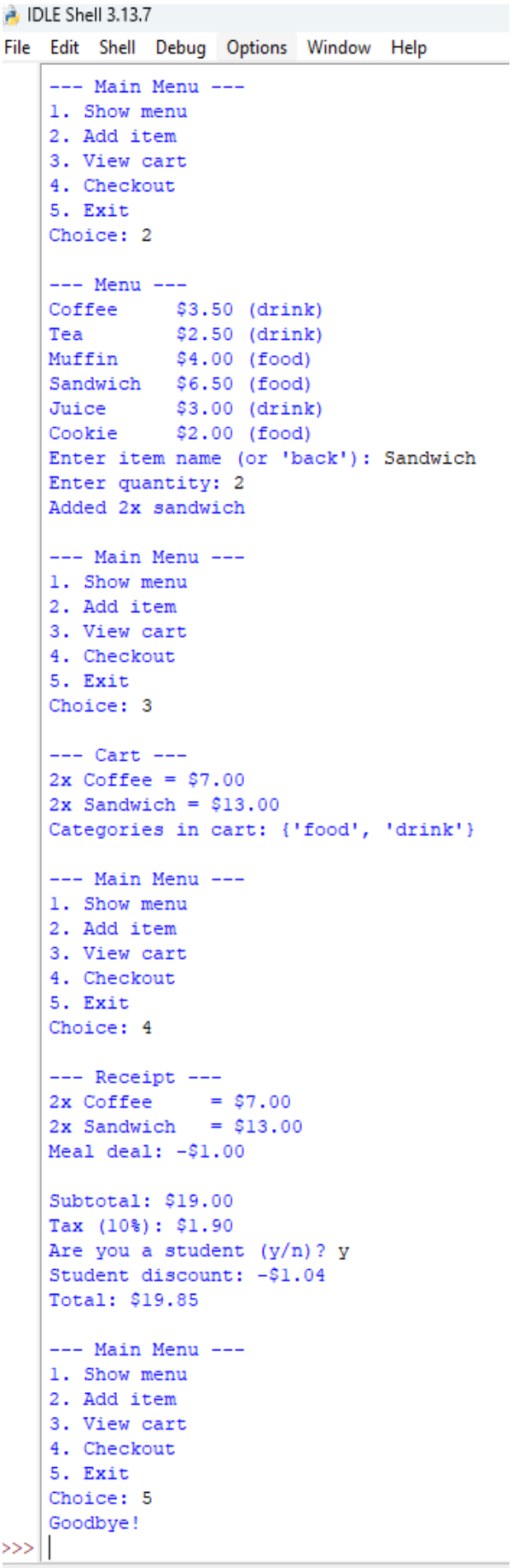
Test Case 3

- Input: View cart empty →
- Expected Outcome: "Cart is empty." PASS.

Test runs:

```
========== RESTART: D:/UC/IIT/Assignments/cafe.py ==========

--- Main Menu ---
1. Show menu
2. Add item
3. View cart
4. Checkout
5. Exit
Choice: 1

--- Menu ---
Coffee      $3.50 (drink)
Tea         $2.50 (drink)
Muffin      $4.00 (food)
Sandwich    $6.50 (food)
Juice       $3.00 (drink)
Cookie      $2.00 (food)

--- Main Menu ---
1. Show menu
2. Add item
3. View cart
4. Checkout
5. Exit
Choice: Coffee
Invalid option.

--- Main Menu ---
1. Show menu
2. Add item
3. View cart
4. Checkout
5. Exit
Choice: 1

--- Menu ---
Coffee      $3.50 (drink)
Tea         $2.50 (drink)
Muffin      $4.00 (food)
Sandwich    $6.50 (food)
Juice       $3.00 (drink)
Cookie      $2.00 (food)

--- Main Menu ---
1. Show menu
2. Add item
3. View cart
4. Checkout
5. Exit
Choice: 2

--- Menu ---
Coffee      $3.50 (drink)
Tea         $2.50 (drink)
Muffin      $4.00 (food)
Sandwich    $6.50 (food)
Juice       $3.00 (drink)
```


IDLE Shell 3.13.7
File Edit Shell Debug Options Window Help
2. Add item
3. View cart
4. Checkout
5. Exit
Choice: 2
--- Menu ---
Coffee $3.50 (drink)
Tea $2.50 (drink)
Muffin $4.00 (food)
Sandwich $6.50 (food)
Juice $3.00 (drink)
Cookie $2.00 (food)
Enter item name (or 'back'): Coffee
Enter quantity: 2
Added 2x coffee
--- Main Menu ---
1. Show menu
2. Add item
3. View cart
4. Checkout
5. Exit
Choice: 1
--- Menu ---
Coffee $3.50 (drink)
Tea $2.50 (drink)
Muffin $4.00 (food)
Sandwich $6.50 (food)
Juice $3.00 (drink)
Cookie $2.00 (food)
--- Main Menu ---
1. Show menu
2. Add item
3. View cart
4. Checkout
5. Exit
Choice: Sandwich
Invalid option.
--- Main Menu ---
1. Show menu
2. Add item
3. View cart
4. Checkout
5. Exit
Choice: 2
--- Menu ---
Coffee $3.50 (drink)
Tea $2.50 (drink)
Muffin $4.00 (food)
Sandwich $6.50 (food)
Juice $3.00 (drink)
Cookie $2.00 (food)
Enter item name (or 'back'): Sandwich
Enter quantity: 2
Added 2x sandwich


IDLE Shell 3.13.7
File Edit Shell Debug Options Window Help
--- Main Menu ---
1. Show menu
2. Add item
3. View cart
4. Checkout
5. Exit
Choice: 2
--- Menu ---
Coffee $3.50 (drink)
Tea $2.50 (drink)
Muffin $4.00 (food)
Sandwich $6.50 (food)
Juice $3.00 (drink)
Cookie $2.00 (food)
Enter item name (or 'back'): Sandwich
Enter quantity: 2
Added 2x sandwich
--- Main Menu ---
1. Show menu
2. Add item
3. View cart
4. Checkout
5. Exit
Choice: 3
--- Cart ---
2x Coffee = $7.00
2x Sandwich = $13.00
Categories in cart: {'food', 'drink'}
--- Main Menu ---
1. Show menu
2. Add item
3. View cart
4. Checkout
5. Exit
Choice: 4
--- Receipt ---
2x Coffee = $7.00
2x Sandwich = $13.00
Meal deal: -$1.00
Subtotal: $19.00
Tax (10%): $1.90
Are you a student (y/n)? y
Student discount: -$1.04
Total: $19.85
--- Main Menu ---
1. Show menu
2. Add item
3. View cart
4. Checkout
5. Exit
Choice: 5
Goodbye!
>>>

→ Hand Written Works for step 06

Cafe.py

Example 01 → • Two Coffees.
1 each $3.50, $3.50 × 2
= $7

• Two Sandwich
1 for each $ $6.50, $6.50 × 2
= $13

• Drink & Food Two categories are there.
So Apply the Meal Deal Discount

= $7 + $13

= [redacted] $20

= $20 - $1 (meal deal)

= [redacted] $19

• Already student then $1.04 discount have

• $19 - $1.04

Total = Total 19.85

Step 7 — Refinement with GenAI

Prompt used:
"can I have python code for campus cafe checkout"
GenAI (ChatGPT) gave a complete working solution with main menu, functions, cart, and receipt.

What I changed and justified:

- Identified some input handling (example:-check quantities as int).
- Justified pseudocode and flowchart with rubric steps.
- Added manual test cases with expected results.

Reasons to change: try to align with marking rubric and keep evidence of the overall process.

Case Study 2 — Smart Classroom Monitor

Step 1: Understanding (5 marks)

Restated problem in my own words:
I need to create a Smart Classroom Monitor program that keeps track of:

- The room state (whether the projector is on, the room capacity, and the current topic).
- The attendance list of students (using a set).
- A temperature log (list of floats) with tools to calculate statistics (min, max, avg).

The program should raise alerts if:

- Attendance exceeds capacity → "ROOM FULL"
- Temperature goes below 16°C or above 28°C → warning
- A topic is set but the projector is off when exiting → reminder

The program should use functions and it should have a while loop to provide a user menu.

Step 2: Inputs & Outputs (5 marks)

Inputs:

1. Main Menu choice (integer/string)
2. Student name for check-in/ check-out (str)
3. Temperature reading (float)
4. Topic name (string)
5. Projector toggle (boolean → via "yes/no")

Outputs

1. Current room's state (projector, capacity, topic)
2. Attendance count and list of names which are entered
3. Temperature stats (min, max, avg)

4. Alerts:
    - “ROOM FULL” if over capacity
    - “The temperature is out of range!” if <16 or >28
    - “Reminder: Topic set but projector OFF!”

Step 3: Algorithm Design (7.5 marks)

Task Breakdown:

1. Initializing the structures:
    - Room dict = { "projector_on": False, "capacity": 3, "topic": "" }
    - Attendance = Empty set
    - Temperature log = empty list
2. Menu loop:

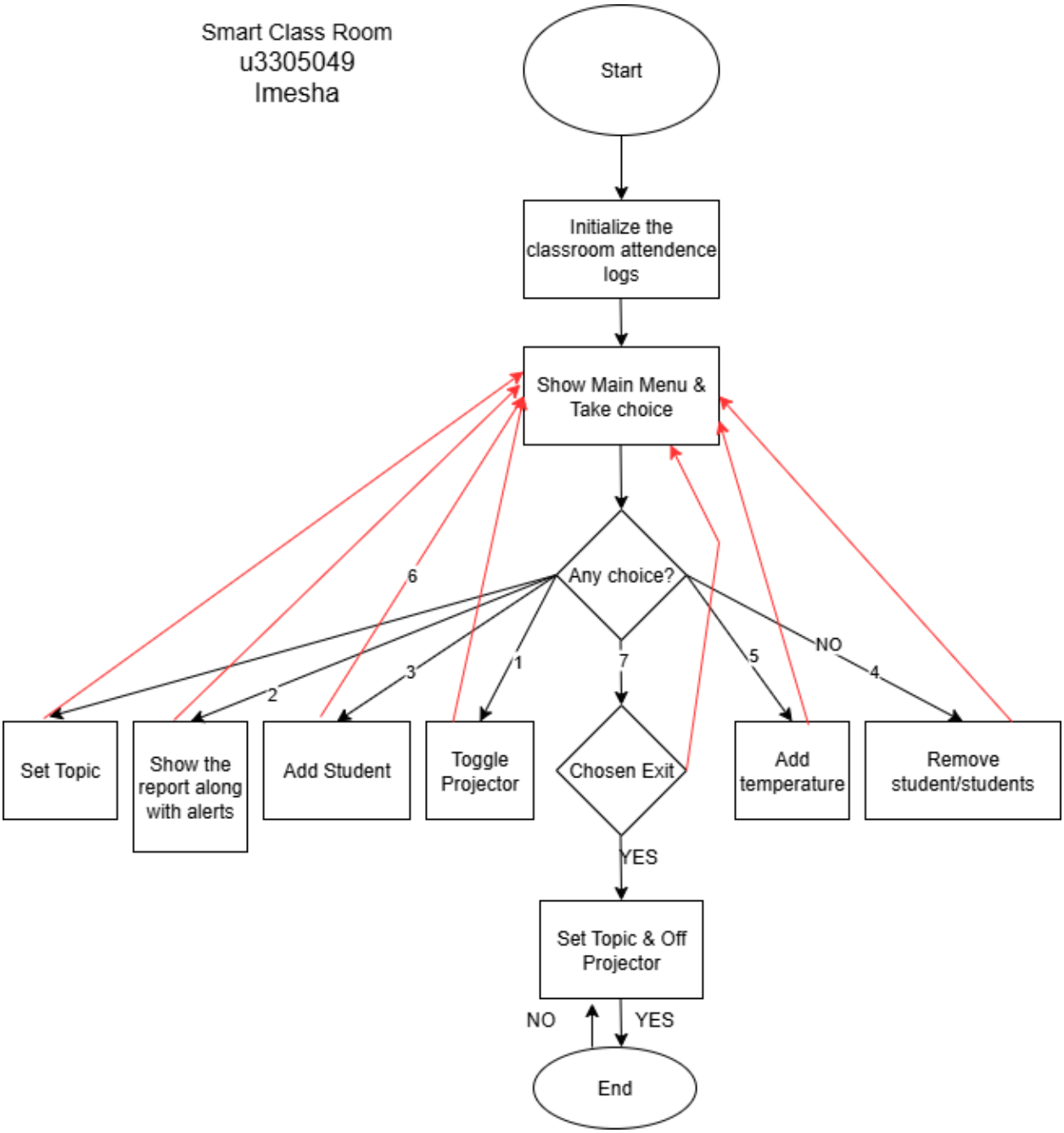
- (1) Toggle the projector
- (2) Set a topic
- (3) Add a student
- (4) Remove the student
- (5) Add the temperature reading
- (6) Show the report (state, attendance, stats, alerts)
- (7) Exit

3. **Functions:**

- toggle_projector(room) → flips the projector state
- set_topic(room) → assigns a string to topic
- add_student(attendance, room) → adds all the names, checks capacity
- remove_student(attendance) → remove the names if present
- add_temp(log) → adds float to the list
- temp_stats(log) → returns (min, max, avg) tuple
- report(room, attendance, log) → prints state, stats and alerts

4. **Exit checking:**
   If exiting and topic is set but the projector is OFF → print reminder.

Step 4: Flowchart (5 marks)



draw.io link:

https://drive.google.com/file/d/17-PwQ2exSDHxRjpuHff_U9Dfs3XXd6p7/view?usp=sharing

Pseudocode for Smart Classroom Monitor

```
room = {"projector_on": False, "capacity": 3, "topic": ""}
attendance = set()
temps = []

while True:
    show entire menu
    choice = input()
    if choice == 1: toggle_projector(room)
    elif choice == 2: set_a_topic(room)
    elif choice == 3: add_a_student(attendance, room)
    elif choice == 4: remove_the_student(attendance)
    elif choice == 5: add_temperature(temps)
    elif choice == 6: Whole_report(room, attendance, temps)
    elif choice == 7:
        if room["topic"] != "" and not room["projector_on"]:
            print reminder
        Break
```

Step 5: Python Implementation (15 marks)

```
# Smart Classroom Monitor
# Case Study 2
# Room state
#u3305049
#Imesha

room = {
   "projector_on": False,
   "capacity": 3,
   "topic": ""
}

attendance = set()
temps = []


def toggle_projector(room):
   room["projector_on"] = not room["projector_on"]
   state = "ON" if room["projector_on"] else "OFF"
   print(f"Projector is now {state}")


def set_topic(room):
   topic = input("Enter topic: ")
   room["topic"] = topic
   print(f"Topic set to: {topic}")

#Add Students
def add_student(attendance, room):
   name = input("Enter student's name: ")
   attendance.add(name)
   print(f"{name} checked in.")
   if len(attendance) > room["capacity"]:
      print(" ⚠️ ALERT: ROOM FULL!")

#Remove Students
def remove_student(attendance):
   name = input("Enter student's name to remove: ")
   if name in attendance:
```

```
        attendance.remove(name)
        print(f"{name} checked out.")
    else:
        print("Student is not found.")

#Add temperature
def add_temp(temps):
    try:
        t = float(input("Enter Temperature (°C): "))
        temps.append(t)
        if t < 16 or t > 28:
            print(" ⚠️ ALERT: Temperature is out of range!")
    except ValueError:
        print("Invalid Temperature!")


def temp_stats(temps):
    if not temps:
        return None
    return (min(temps), max(temps), sum(temps) / len(temps))


def report(room, attendance, temps):
    print("\n--- Classroom Report ---")
    print(f"Projector: {'ON' if room['projector_on'] else 'OFF'}")
    print(f"Capacity: {len(attendance)}/{room['capacity']}")
    print(f"Topic: {room['topic'] if room['topic'] else 'None'}")
    print(f"Students: {attendance}")
    stats = temp_stats(temps)
    if stats:
        print(f"Temperatures → Min: {stats[0]:.1f}, Max: {stats[1]:.1f}, Avg: {stats[2]:.1f}")
    else:
        print("No temperature logs yet.")

#Main Menu
def main():
    while True:
        print("\n--- Main Menu ---")
        print("1. Toggle projector")
        print("2. Set topic")
        print("3. Add student")
```

```
        print("4. Remove student")
        print("5. Add temperature reading")
        print("6. Show report")
        print("7. Exit")
        choice = input("Choice: ")

        if choice == "1":
            toggle_projector(room)
        elif choice == "2":
            set_topic(room)
        elif choice == "3":
            add_student(attendance, room)
        elif choice == "4":
            remove_student(attendance)
        elif choice == "5":
            add_temp(temps)
        elif choice == "6":
            report(room, attendance, temps)
        elif choice == "7":
            if room["topic"] and not room["projector_on"]:
                print("Reminder1: Topic is set but the projector is OFF!")
            print("Exiting...")
            break
        else:
            print("Invalid option!")


if __name__ == "__main__":
    main()
```

Test Results:


IDLE Shell 3.13.7
File Edit Shell Debug Options Window Help
7. Exit
Choice: 3
Enter student name: Imesha
Imesha checked in.
--- Main Menu ---
1. Toggle projector
2. Set topic
3. Add student
4. Remove student
5. Add temperature reading
6. Show report
7. Exit
Choice: 5
Enter temperature (°C): 15
⚠ ALERT: Temperature out of range!
--- Main Menu ---
1. Toggle projector
2. Set topic
3. Add student
4. Remove student
5. Add temperature reading
6. Show report
7. Exit
Choice: 6
--- Classroom Report ---
Projector: ON
Capacity: 1/3
Topic: 3
Students: {'Imesha'}
Temperatures → Min: 15.0, Max: 15.0, Avg: 15.0
--- Main Menu ---
1. Toggle projector
2. Set topic
3. Add student
4. Remove student
5. Add temperature reading
6. Show report
7. Exit
Choice: 7
Exiting...
>>>
= RESTART: D:/UC/IIT/Assignments/Smart Classroom Monitor_Python Implementation.py

Step 6: Testing (5 marks)

Test Case 1

Input: Add 4 students to system, (capacity = 3)

Expected Outcome: After entering 4th → showed “ROOM FULL!” alert

Test Case 2

Input: Add temperatures to 20, 15, 30

Expected Outcome: Should receive-range-of-alerts for 15 and 30. Stats = Min=15, Max=30, Avg=21.7

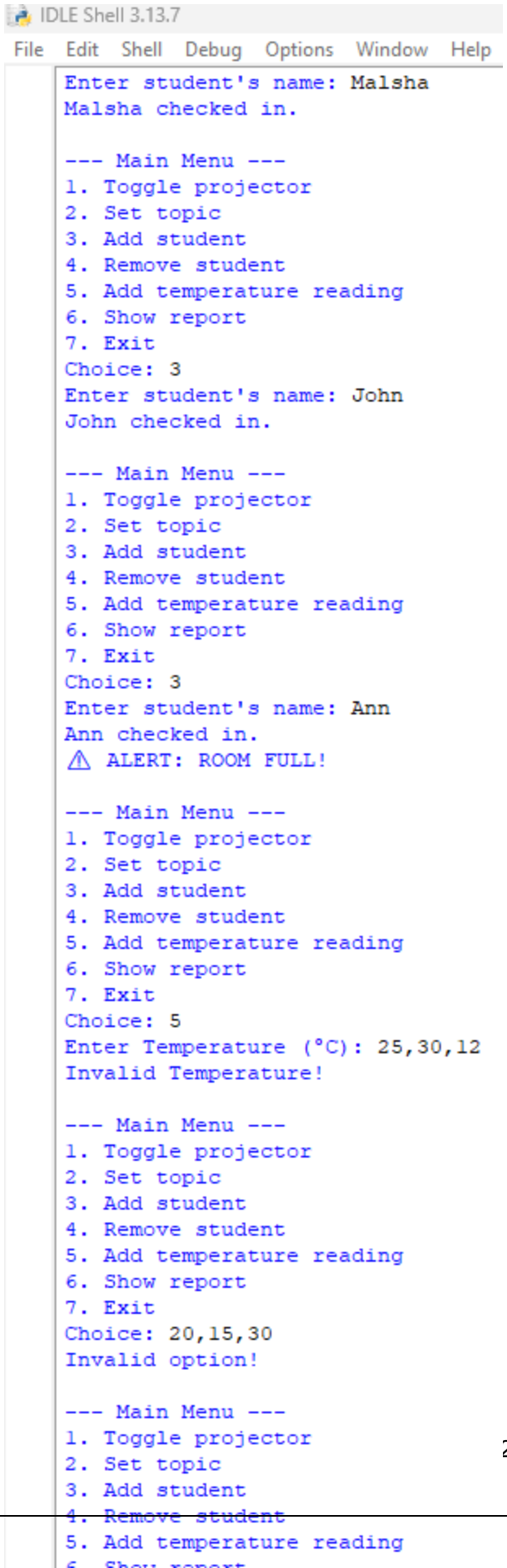
Test Case 3

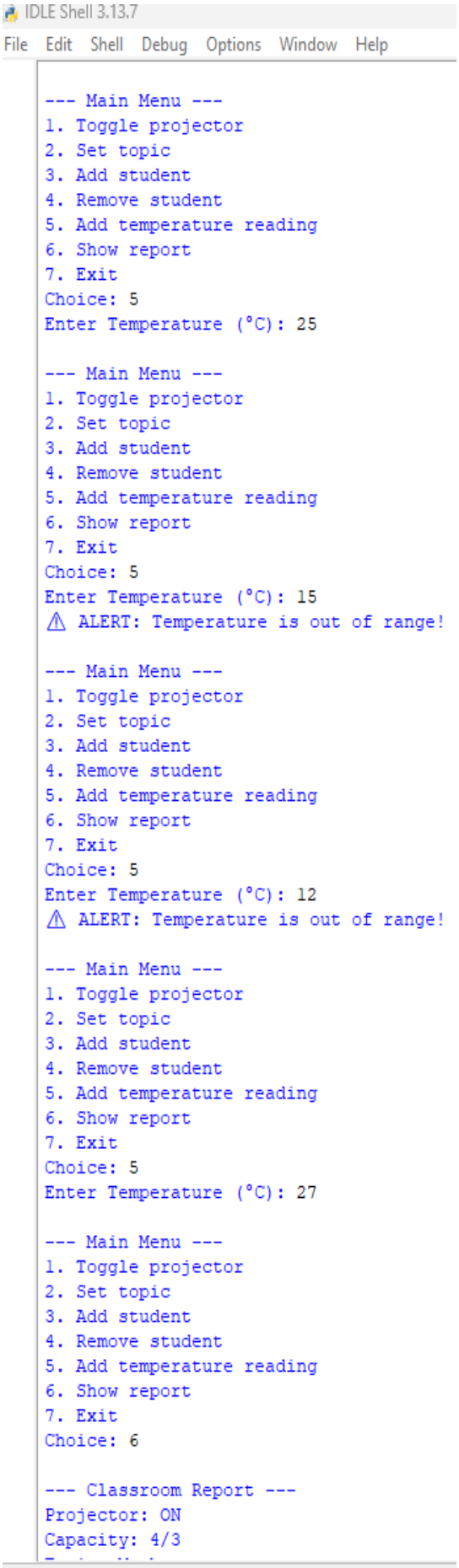
Input: Set topic = “Math”, exit with projector OFF

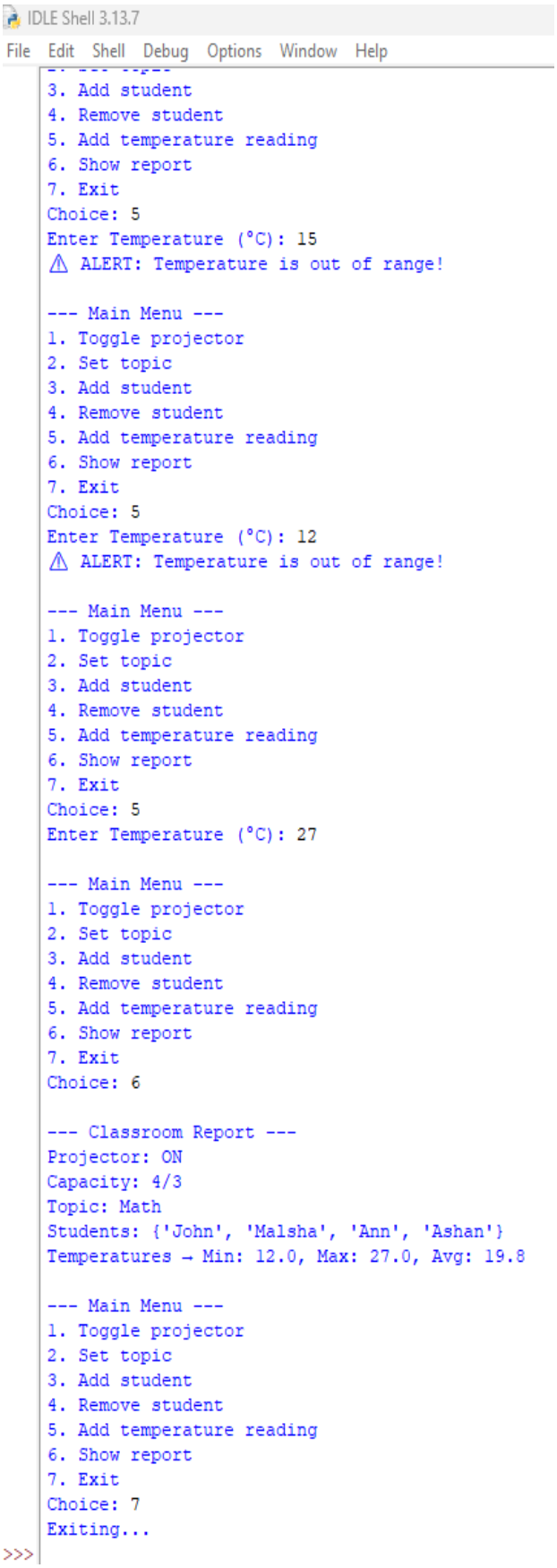
Expected Outcome: Reminder the message

Some works:

```
>>>
====== RESTART: D:/UC/IIT/Assignments/Smart Classroom Monitor_Python Implementation.py ======

--- Main Menu ---
1. Toggle projector
2. Set topic
3. Add student
4. Remove student
5. Add temperature reading
6. Show report
7. Exit
Choice: 1
Projector is now ON

--- Main Menu ---
1. Toggle projector
2. Set topic
3. Add student
4. Remove student
5. Add temperature reading
6. Show report
7. Exit
Choice: 2
Enter topic: Math
Topic set to: Math

--- Main Menu ---
1. Toggle projector
2. Set topic
3. Add student
4. Remove student
5. Add temperature reading
6. Show report
7. Exit
Choice: 3
Enter student's name: Ashan
Ashan checked in.

--- Main Menu ---
1. Toggle projector
2. Set topic
3. Add student
4. Remove student
5. Add temperature reading
6. Show report
7. Exit
Choice: 3
Enter student's name: Malsha
Malsha checked in.

--- Main Menu ---
1. Toggle projector
2. Set topic
3. Add student
4. Remove student
5. Add temperature reading
6. Show report
```


IDLE Shell 3.13.7
File Edit Shell Debug Options Window Help
Enter student's name: Malsha
Malsha checked in.
--- Main Menu ---
1. Toggle projector
2. Set topic
3. Add student
4. Remove student
5. Add temperature reading
6. Show report
7. Exit
Choice: 3
Enter student's name: John
John checked in.
--- Main Menu ---
1. Toggle projector
2. Set topic
3. Add student
4. Remove student
5. Add temperature reading
6. Show report
7. Exit
Choice: 3
Enter student's name: Ann
Ann checked in.
⚠ ALERT: ROOM FULL!
--- Main Menu ---
1. Toggle projector
2. Set topic
3. Add student
4. Remove student
5. Add temperature reading
6. Show report
7. Exit
Choice: 5
Enter Temperature (°C): 25,30,12
Invalid Temperature!
--- Main Menu ---
1. Toggle projector
2. Set topic
3. Add student
4. Remove student
5. Add temperature reading
6. Show report
7. Exit
Choice: 20,15,30
Invalid option!
--- Main Menu ---
1. Toggle projector
2. Set topic
3. Add student
4. Remove student
5. Add temperature reading


IDLE Shell 3.13.7
File Edit Shell Debug Options Window Help
--- Main Menu ---
1. Toggle projector
2. Set topic
3. Add student
4. Remove student
5. Add temperature reading
6. Show report
7. Exit
Choice: 5
Enter Temperature (°C): 25
--- Main Menu ---
1. Toggle projector
2. Set topic
3. Add student
4. Remove student
5. Add temperature reading
6. Show report
7. Exit
Choice: 5
Enter Temperature (°C): 15
⚠ ALERT: Temperature is out of range!
--- Main Menu ---
1. Toggle projector
2. Set topic
3. Add student
4. Remove student
5. Add temperature reading
6. Show report
7. Exit
Choice: 5
Enter Temperature (°C): 12
⚠ ALERT: Temperature is out of range!
--- Main Menu ---
1. Toggle projector
2. Set topic
3. Add student
4. Remove student
5. Add temperature reading
6. Show report
7. Exit
Choice: 5
Enter Temperature (°C): 27
--- Main Menu ---
1. Toggle projector
2. Set topic
3. Add student
4. Remove student
5. Add temperature reading
6. Show report
7. Exit
Choice: 6
--- Classroom Report ---
Projector: ON
Capacity: 4/3


IDLE Shell 3.13.7
File Edit Shell Debug Options Window Help
3. Add student
4. Remove student
5. Add temperature reading
6. Show report
7. Exit
Choice: 5
Enter Temperature (°C): 15
⚠ ALERT: Temperature is out of range!
--- Main Menu ---
1. Toggle projector
2. Set topic
3. Add student
4. Remove student
5. Add temperature reading
6. Show report
7. Exit
Choice: 5
Enter Temperature (°C): 12
⚠ ALERT: Temperature is out of range!
--- Main Menu ---
1. Toggle projector
2. Set topic
3. Add student
4. Remove student
5. Add temperature reading
6. Show report
7. Exit
Choice: 5
Enter Temperature (°C): 27
--- Main Menu ---
1. Toggle projector
2. Set topic
3. Add student
4. Remove student
5. Add temperature reading
6. Show report
7. Exit
Choice: 6
--- Classroom Report ---
Projector: ON
Capacity: 4/3
Topic: Math
Students: {'John', 'Malsha', 'Ann', 'Ashan'}
Temperatures → Min: 12.0, Max: 27.0, Avg: 19.8
--- Main Menu ---
1. Toggle projector
2. Set topic
3. Add student
4. Remove student
5. Add temperature reading
6. Show report
7. Exit
Choice: 7
Exiting...
>>>

Smart Classroom Monitor

U3305049

Imesha

(*) → Temperature is between t<16 or t>28

t means temperature.

→ When we are adding temperatur = 12 It is being error. because of the temperature range.

12 is lower than 16.

(*) We can add students Maximum 3. If we are trying to add more than 3 students. It is received anerror!

Step 7: Refinement via GenAI (5 marks)

Prompt used:
"In Case Study 02 — Smart Classroom Monitor … can I have an answer for this question with rubric?"

What is changed and justification:

- GenAI is a structured approach with functions.
- It simplified input validation for temperatures.
- Added a clear alert icons ⚠️ for to easy to visible.
- Created a report function to show all main key details in one place.

Reason to do: Keeping code clean, easy to read, and aligned with the rubric requirements.

Case Study 3 — Boolean Circuit Equivalence

01.

a) A'

b) C`

c) (A` * C`)`

d) B`

e) (A` * C`)` * A * B`

f) A * C * B

g) (A * C * B) + ((A` * C`)` * A * B`)

h) B`

i) B` + C

j) A * (B` + C)

Extra Work

According to my knowledge I simplified a,b,c,d,e,f,g more like below.

a) = A'

b) = C`

c) = (A` * C`)` = (A + C)`` = (A + C)

d) = B`

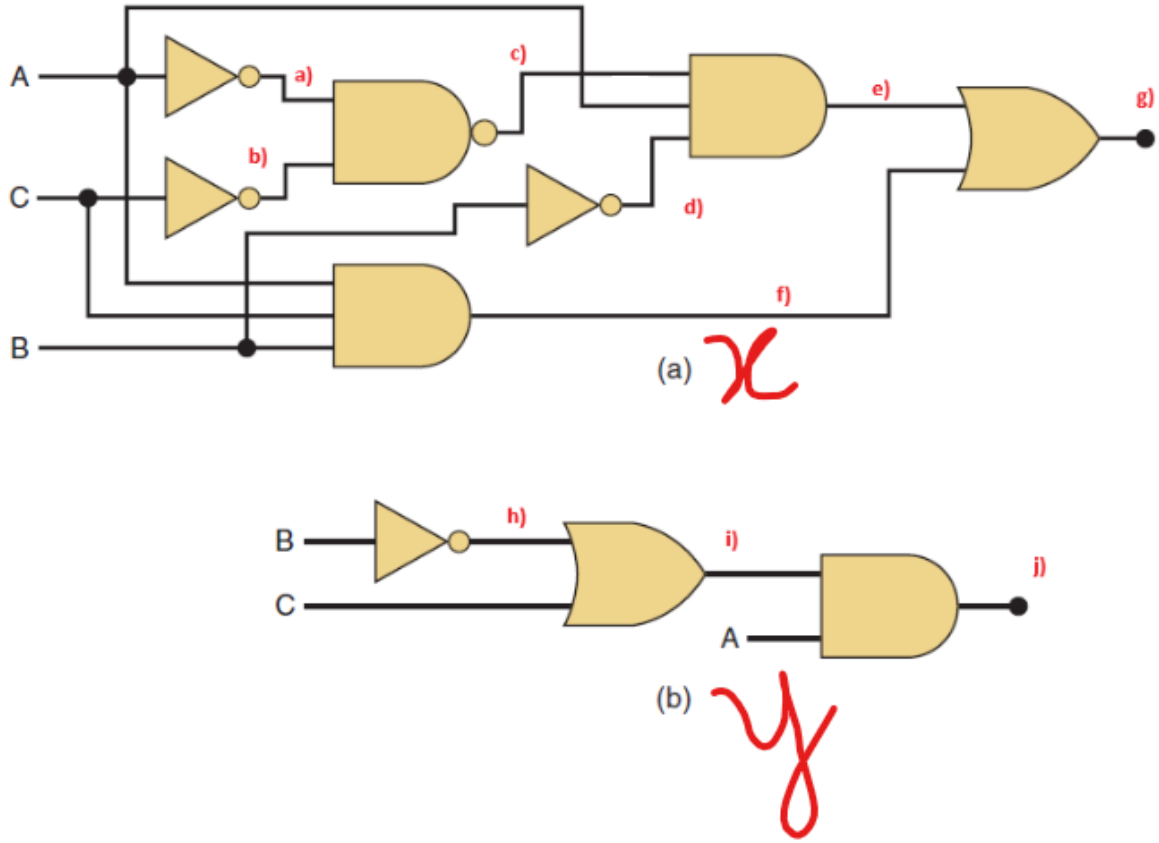
e) = (A + C) * A * B` = A * B`

f) = A * C * B

g) = (A * C * B) + (A * B`)

02.python code

Figure Case study 3:



Student: Imesha Piyumali, ID: u3305049, Date: 17.09.2025

#Case Study 3 — Boolean Circuit Equivalence

#IMPORTANT NOTICE X= a CIRCUIT

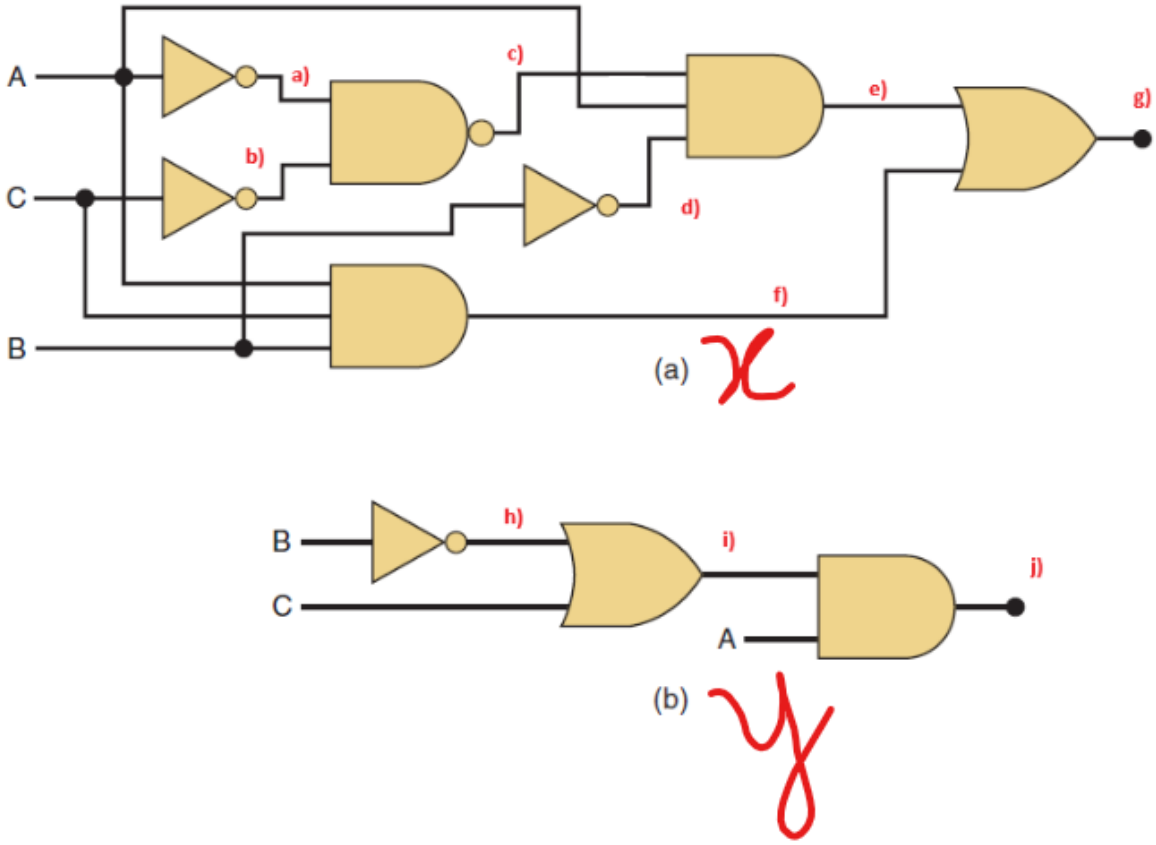
#Y= b CIRCUIT

```
#Circuit Outputs
def circuit_outputs(A: int, B: int, C: int):
    # Complements
    A_ = 1 - A
    B_ = 1 - B
    C_ = 1 - C



#IMPORTANT NOTICE X= a CIRCUIT
#Y= b CIRCUIT



    # Circuit X: (A * C * B) + ((A' * C')' * A * B')
    part1 = A and C and B
    part2 = (not (A_ and C_)) and A and B_
    X = int(part1 or part2)

    # Circuit Y: A * (B' + C)
    Y = int(A and (B_ or C))

    return X, Y
#IMPORTANT NOTICE X= a CIRCUIT
#Y= b CIRCUIT
```

```
if __name__ == "__main__":
    print("Enter values for A, B, C (0 or 1). Type 'q' to quit.")
    while True:
        user_input = input("A B C: ").strip()
        if user_input.lower() == "q":
            print("Exiting!!!")
            break
        try:
            A_str, B_str, C_str = user_input.split()
            A, B, C = int(A_str), int(B_str), int(C_str)
            if A not in [0,1] or B not in [0,1] or C not in [0,1]:
                raise ValueError
            X, Y = circuit_outputs(A, B, C)
            print(f"Inputs: A={A}, B={B}, C={C} => X={X}, Y={Y}")
        except ValueError:
            print("Invalid Input here. Please enter three values (0 or 1), separated with spaces.")


#IMPORTANT NOTICE X= a CIRCUIT
#Y= b CIRCUIT
```

Test Results:

Figure Case study 3:



#IMPORTANT NOTICE

X= a CIRCUIT

#Y= b CIRCUIT

```
>>>
=============== RESTART: D:/UC/IIT/Assignments/Case Study 3_A Boolean Circuit Equivalence.py ===============
Enter values for A, B, C (0 or 1). Type 'q' to quit.
A B C:  0 0 0
Inputs: A=0, B=0, C=0 => X=0, Y=0
A B C:  0 0 1
Inputs: A=0, B=0, C=1 => X=0, Y=0
A B C:  0 1 0
Inputs: A=0, B=1, C=0 => X=0, Y=0
A B C:  0 1 1
Inputs: A=0, B=1, C=1 => X=0, Y=0
A B C:  1 0 0
Inputs: A=1, B=0, C=0 => X=1, Y=1
A B C:  1 0 1
Inputs: A=1, B=0, C=1 => X=1, Y=1
A B C:  1 1 0
Inputs: A=1, B=1, C=0 => X=0, Y=0
A B C:  1 1 1
Inputs: A=1, B=1, C=1 => X=1, Y=1
A B C:  0 0 0
Inputs: A=0, B=0, C=0 => X=0, Y=0
A B C:
```

According to the test results, X(a circuit) and Y(b circuit) output are always equal. So it means a and b circuits are equivalent.

03. Truth Tables

(a)

A	B	C	A'	C'	(A'*C')	(A'*C')'	(A*C*B)	((A'*C')'*A*B')	F
0	0	0	1	1	1	0	0	0	0
0	0	1	1	0	0	1	0	0	0
0	1	0	1	1	1	0	0	0	0
0	1	1	1	0	0	1	0	0	0
1	0	0	0	1	0	1	0	1	1
1	0	1	0	0	0	1	0	1	1
1	1	0	0	1	0	1	0	0	0
1	1	1	0	0	0	1	1	0	1

(b)

A	B	C	B'	(B'+C)	F
0	0	0	1	1	0
0	0	1	1	1	0
0	1	0	0	0	0
0	1	1	0	1	0
1	0	0	1	1	1
1	0	1	1	1	1
1	1	0	0	0	0
1	1	1	0	1	1